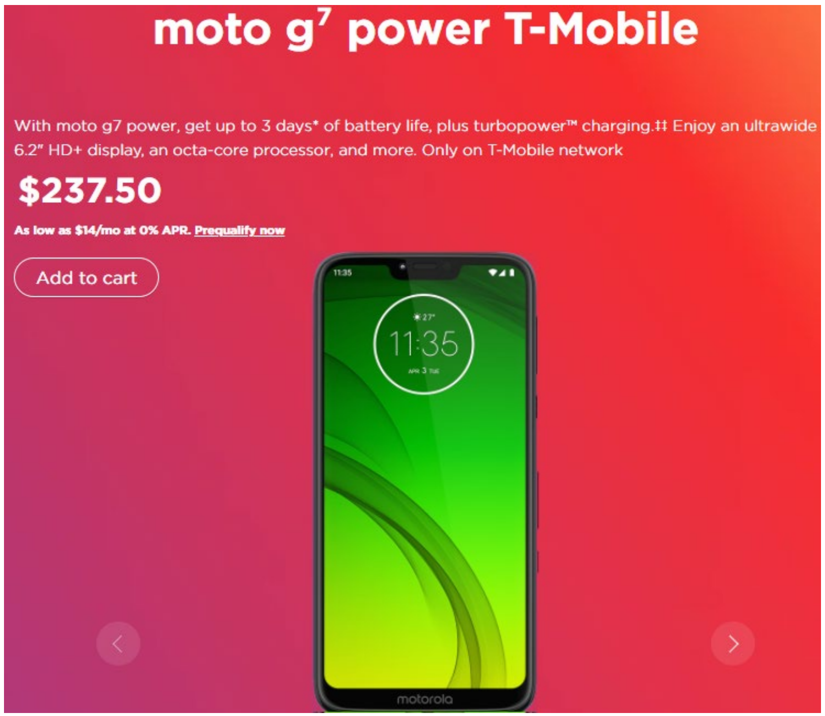
Exhibit E

Claim Chart – '532 Patent

US Patent 6,473,532 Versus Moto G7 Power Smartphone Video Recording



International Telecommunication Union

ITU-T

TELECOMMUNICATION STANDARDIZATION SECTOR OF ITU

H.264
(11/2007)

SERIES H: AUDIOVISUAL AND MULTIMEDIA SYSTEMS

Infrastructure of audiovisual services – Coding of moving video

Advanced Video Coding

From Wikipedia, the free encyclopedia

"AVC1" redirects here. It is not to be confused with AV1 or VC-1.

Advanced Video Coding (**AVC**), also referred to as **H.264** or **MPEG-4 Part 10, Advanced Video Coding** (**MPEG-4 AVC**), is a video compression standard based on block-oriented, motion-compensated integer-DCT coding.[1] It is by far the most commonly used format for the recording, compression, and distribution of video content, used by 91% of video industry developers as of September 2019.[2][3][4] It supports resolutions up to and including 8K UHD.[5][6]

6. A method of visual lossless encoding of frames of a video signal, the method comprising steps of:

spatially and temporally separating and analyzing details of said frames;

estimating parameters of said details;

defining a visual perception threshold for each of said details in accordance with said estimated detail parameters;

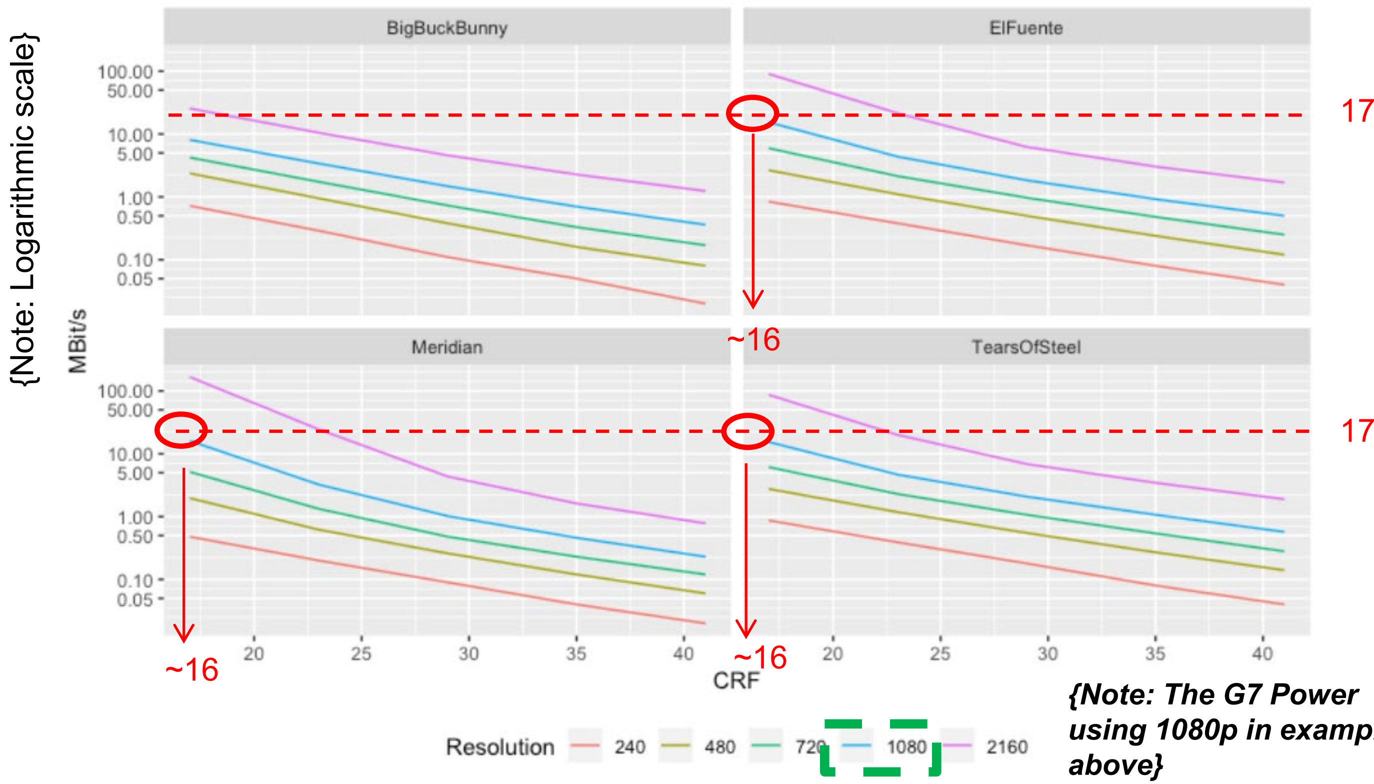
classifying said frame picture details into subclasses in accordance with said visual perception thresholds and said detail parameters; and

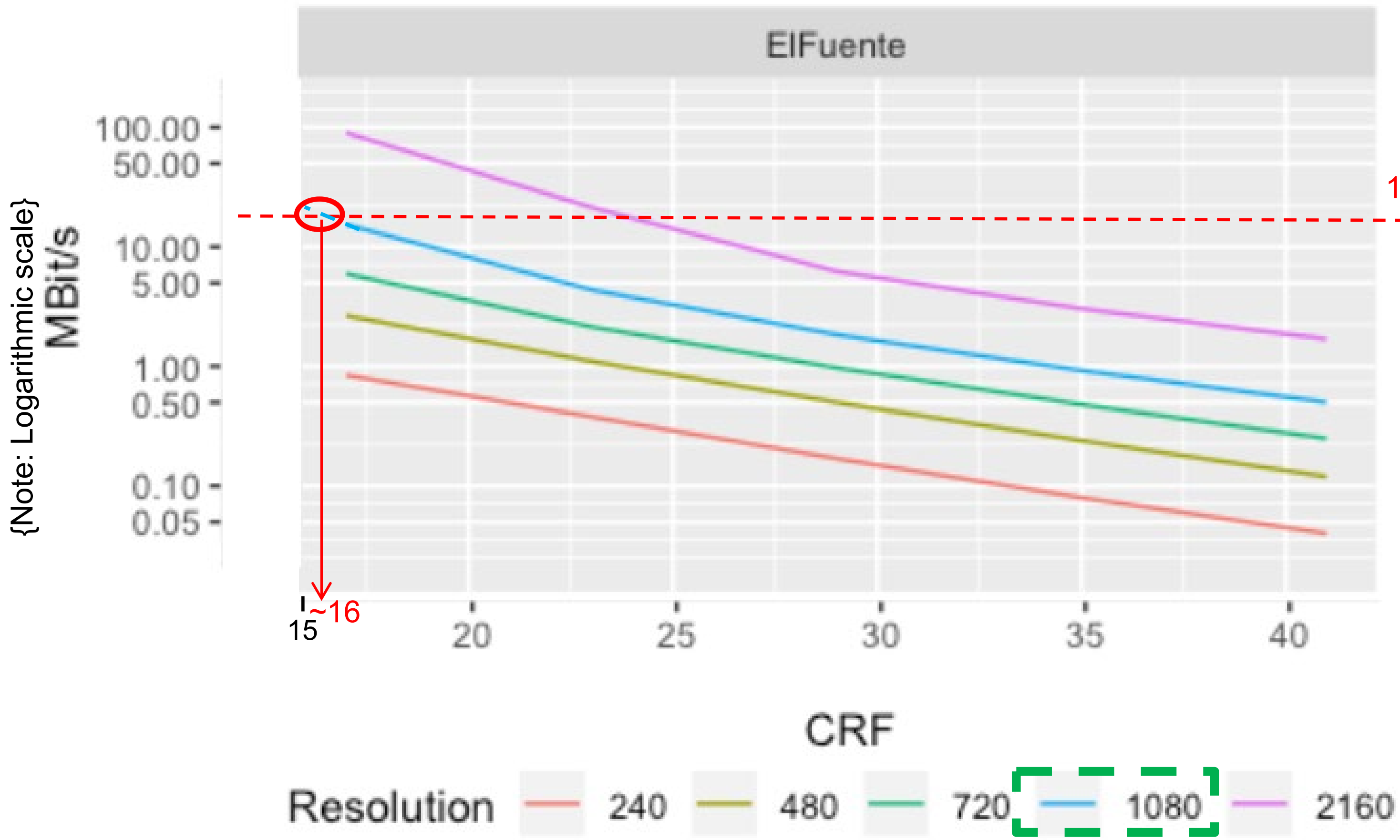
transforming each said frame detail in accordance with its associate subclass.

Preliminary Claim Chart Showing Infringement of Claim 6 of the U.S. Patent No. 6,473,532 by Lenovo

Claim 6	
A method of visual lossless encoding of frames of a video signal, the method comprising steps of:	https://www.gsmarena.com/motorola_moto_g7_power-review-1889p5.php  … Videos shot on the Motorola G7 Power in 4K and 1080p resolution at 30 fps get saved in a rather standard configuration of a 17-ish Mbps AVC video feed and a 48kHz stereo AAC audio track, inside an MP4 container. The frame rate remains pretty steady at 30 fps. Quality is actually quite good with plenty of detail for the class, high contrast, and lively colors. The dynamic range is about average.

Preliminary Claim Chart Showing Infringement of Claim 6 of the U.S. Patent No. 6,473,532 by Lenovo

Claim 6	
A method of visual lossless encoding of frames of a video signal, the method comprising steps of:	https://slhck.info/video/2017/02/24/crf-guide.html ***{Note: G7 Power video is encoded with "Constant Rate Factor" (CRF) of ~16, from bit rate of of 17 Mpbs and 1080p.}*** **What bitrates will I get?** To give you an estimation of the bitrates to be expected for clips with different resolutions, here's a figure showing the average bitrate in MBit/s for four one-minute video clips with different encoding complexity, encoded with x264 and different CRF values:  ***{Note: The G7 Power was using 1080p in example above}***

Claim 6	
A method of visual lossless encoding of frames of a video signal, the method comprising steps of:	https://slhck.info/video/2017/02/24/crf-guide.html ***{Note: H.264 video is encoded with "Constant Rate Factor" (CRF) <u>of ~16</u>, from bit rate of of 17 Mpbs and 1080p, using data from example video "ElFuente" in slide above}*** 

Preliminary Claim Chart Showing Infringement of Claim 6 of the U.S. Patent No. 6,473,532 by Lenovo

Claim 6

A method of visual lossless encoding of frames of a video signal, the method comprising steps of:

https://trac.ffmpeg.org/wiki/Encode/H.264

{Note: Video from Lumix GH3 camera is encoded with "Constant Rate Factor" (CRF) of 16 or better. A CFR of 16 or better is "visually lossless" }

https://trac.ffmpeg.org/wiki/Encode/H.264

FFmpeg

Search

Login | Preferences | Help/Guide | About Trac | Register

Wiki | Timeline | View Tickets | Search | Tags

wiki: Encode / H.264

+6 | Up | Start Page | Index | History

H.264 Video Encoding Guide

This guide focuses on the encoder x264. It assumes you have `ffmpeg` compiled with `--enable-libx264`. If you need help compiling and installing see one of our compiling guides. See HWAccelIntro for information on supported hardware H.264 encoders.

Contents

There are two rate control modes that are usually suggested for general use: Constant Rate Factor (CRF) or Two-Pass ABR. Rate control decides how many bits will be used for each frame. This will determine the file size and also how quality is distributed. To know more about what the different rate control modes do see this post.

Constant Rate Factor (CRF)

Use this rate control mode if you want to keep the best quality and care less about the file size. This is the recommended rate control mode for most uses.

This method allows the encoder to attempt to achieve a certain output quality for the whole file when output file size is of less importance. This provides maximum compression efficiency with a single pass. By adjusting the so-called quantizer for each frame, it gets the bitrate it needs to keep the requested quality level. The downside is that you can't tell it to get a specific filesize or not go over a specific size or bitrate, which means that this method is not recommended for encoding videos for streaming.

1. Choose a CRF value

The range of the CRF scale is 0–51, where 0 is lossless, 23 is the default, and 51 is worst quality possible. A lower value generally leads to higher quality, and a subjectively sane range is 17–28. Consider 17 or 18 to be visually lossless or nearly so; it should look the same or nearly the same as the input but it isn't technically lossless.

The range is exponential, so increasing the CRF value +6 results in roughly half the bitrate / file size, while -6 leads to roughly twice the bitrate.

Choose the highest CRF value that still provides an acceptable quality. If the output looks good, then try a higher value. If it looks bad, choose a lower value.

Preliminary Claim Chart Showing Infringement of Claim 6 of the U.S. Patent No. 6,473,532 by Lenovo

Claim 6

A method of visual lossless encoding of frames of a video signal, the method comprising steps of:

H.264/MPEG-4 AVC is the latest international video coding standard. It was jointly developed by the Video Coding Experts Group (VCEG) of the ITU-T and the Moving Picture Experts Group (MPEG) of ISO/IEC.

In the process of designing the FRExt amendment, the JVT was able to go back and re-examine several prior technical proposals that had not been included in the initial standard due to scheduling constraints. Specifically, these included:

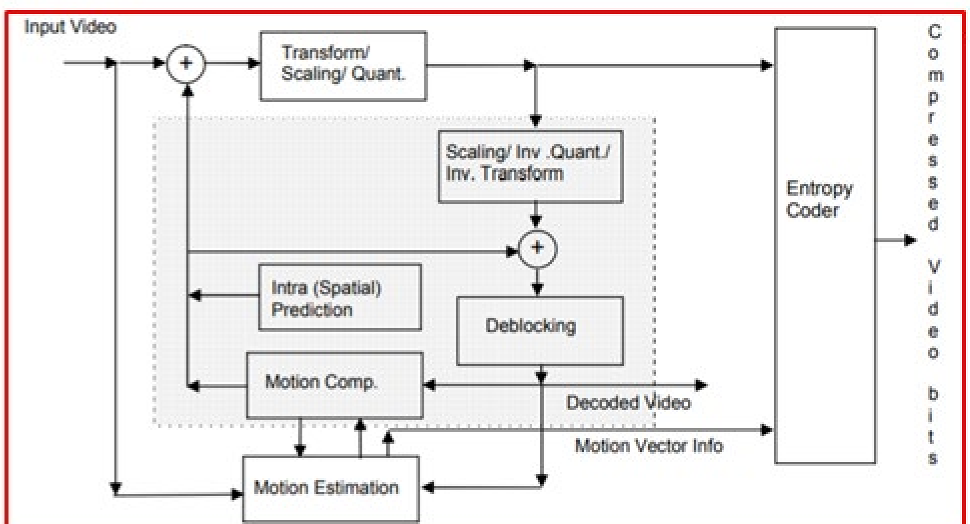
-> Supporting an adaptive block-size for the residual spatial frequency transform,

-> Supporting encoder-specified perceptual-based quantization scaling matrices, and

-> Supporting efficient lossless representation of specific regions in video content.

ITU-T H.264 (06/2019)

The coded representation specified in the syntax is designed to enable a high compression capability for a desired image quality. With the exception of the transform bypass mode of operation for lossless coding in the High 4:4:4 Intra, CAVLC 4:4:4 Intra, and High 4:4:4 Predictive profiles.



http://www.fastvdo.com/spie04/spie04-h264OverviewPaper.pdf

Claim 6	
spatially and temporally separating and analyzing details of said frames;	https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-H.264-200711-S!!PDF-E&type=items pg. 3 **0.6 Overview of the design characteristics** This subclause does not form an integral part of this Recommendation | International Standard. The coded representation specified in the syntax is designed to enable a high compression capability for a desired image quality. With the exception of the transform bypass mode of operation for lossless coding in the High 4:4:4 Intra, CAVLC 4:4:4 Intra, and High 4:4:4 Predictive profiles, and the I_PCM mode of operation in all profiles, the algorithm is typically not lossless, as the exact source sample values are typically not preserved through the encoding and decoding processes. A number of techniques may be used to achieve highly efficient compression. Encoding algorithms (not specified in this Recommendation | International Standard) may select between inter and intra coding for block-shaped regions of each picture. Inter coding uses motion vectors for block-based inter prediction to exploit temporal statistical dependencies between different pictures. Intra coding uses various spatial prediction modes to exploit spatial statistical dependencies in the source signal for a single picture. Motion vectors and intra prediction modes may be specified for a variety of block sizes in the picture. The prediction residual is then further compressed using a transform to remove spatial correlation inside the transform block before it is quantised, producing an irreversible process that typically discards less important visual information while forming a close approximation to the source samples. Finally, the motion vectors or intra prediction modes are combined with the quantised transform coefficient information and encoded using either variable length codes or arithmetic coding.

Preliminary Claim Chart Showing Infringement of Claim 6 of the U.S. Patent No. 6,473,532 by Lenovo

Claim 6	
estimating parameters of said details;	https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-H.264-200711-S!!PDF-E&type=items pg. 11 **3.133 sequence parameter set**: A *syntax structure* containing *syntax elements* that apply to zero or more entire *coded video sequences* as determined by the content of a seq_parameter_set_id *syntax element* found in the *picture parameter set* referred to by the pic_parameter_set_id *syntax element* found in each *slice header*. **7.3.2.1.1 Sequence parameter set data syntax**

seq_parameter_set_data( ) {	C	Descriptor
profile_idc	0	u(8)
constraint_set0_flag	0	u(1)
constraint_set1_flag	0	u(1)
constraint_set2_flag	0	u(1)
constraint_set3_flag	0	u(1)
reserved_zero_4bits /* equal to 0 */	0	u(4)
level_idc	0	u(8)
seq_parameter_set_id	0	ue(v)
if(profile_idc = = 100 || profile_idc = = 110 || profile_idc = = 122 || profile_idc = = 244 || profile_idc = = 44 || profile_idc = = 83 || profile_idc = = 86) {		
chroma_format_idc	0	ue(v)
if(chroma_format_idc = = 3)		
separate_colour_plane_flag	0	u(1)
bit_depth_luma_minus8	0	ue(v)
bit_depth_chroma_minus8	0	ue(v)

Pg. 40

{Note: "High Profile", and others}

Claim 1	
estimating parameters of said details;	G7 Power records with H.264 and "High Profile": https://www.phonearena.com/phones/Motorola-Moto-G-Power_id11349 **CAMERA** ... **Video recording:** 3840x2160 (4K UHD) (30 fps), 1920x1080 (Full HD) (60 fps) "High Definition" uses the H.264 "High Profile": https://www.rgb.com/h264-profiles High Profile H.264 High Profile is the most efficient and powerful profile in the H.264 family, and is the primary profile for broadcast and disc storage, particularly for HDTV and Bluray disc storage formats. It can achieve a compression ratio of about 2000:1. The High Profile also uses an adaptive transform that can select between 4x4 or 8x8-pixel blocks. For example, 4x4 blocks are used for portions of the picture that are dense with detail, while portions that have little detail are compressed using 8x8 blocks. The result is the preservation of video image quality while reducing network bandwidth requirements by up to 50 percent. By applying H.264 High Profile compression, a 1 Gbps stream can be compressed to about 512 Kbps.

Claim 1	
estimating parameters of said details;	https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-H.264-200711-S!!PDF-E&type=items pg. 286 The "High Profile" includes the use of bit_depth_luma and bit_depth_chroma: **A.2.4 High profile** Bitstreams conforming to the High profile shall obey the following constraints: – Only I, P, and B slice types may be present. – NAL unit streams shall not contain nal_unit_type values in the range of 2 to 4, inclusive. – Arbitrary slice order is not allowed. – Picture parameter sets shall have num_slice_groups_minus1 equal to 0 only. – Picture parameter sets shall have redundant_pic_cnt_present_flag equal to 0 only. – Sequence parameter sets shall have chroma_format_idc in the range of 0 to 1 inclusive. – Sequence parameter sets shall have bit_depth_luma_minus8 equal to 0 only. – Sequence parameter sets shall have bit_depth_chroma_minus8 equal to 0 only.

Preliminary Claim Chart Showing Infringement of Claim 6 of the U.S. Patent No. 6,473,532 by Lenovo

<table>
<tr><th>Claim 6</th><th></th></tr>
<tr><td>defining a visual perception threshold for each of said details in accordance with said estimated detail parameters;</td><td>

https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-H.264-200711-S!!PDF-E&type=items

Pg. 201 - 202

8.7.2.2 Derivation process for the thresholds for each block edge

Inputs to this process are:

– the input sample values p_0, q_0, p_1 and q_1 of a single set of samples across an edge that is to be filtered,

– the variables chromaEdgeFlag and bS, for the set of input samples, as specified in clause 8.7.2,

– the variables filterOffsetA, filterOffsetB, qP_p, and qP_q.

Outputs of this process are the variable filterSamplesFlag, which indicates whether the input samples are filtered, the value of indexA, and the values of the threshold variables α and β.

The variables α′ and β′ depending on the values of indexA and indexB are specified in Table 8-16. Depending on chromaEdgeFlag, the corresponding threshold variables α and β are derived as follows:

– If chromaEdgeFlag is equal to 0,

$$\alpha = \alpha' * (1 << (BitDepth_Y - 8)) \quad (8\text{-}456)$$

$$\beta = \beta' * (1 << (BitDepth_Y - 8)) \quad (8\text{-}457)$$

– Otherwise (chromaEdgeFlag is equal to 1),

$$\alpha = \alpha' * (1 << (BitDepth_C - 8)) \quad (8\text{-}458)$$

$$\beta = \beta' * (1 << (BitDepth_C - 8)) \quad (8\text{-}459)$$

{Note: The threshold values of alpha and beta are based on bit_depth parameters for both chromacity (symbolized by c) and luminance (symbolized by y)). }

The variable filterSamplesFlag is derived by

$$filterSamplesFlag = (bS \mathrel{!=} 0 \ \&\& \ Abs(p_0 - q_0) < \alpha \ \&\& \ Abs(p_1 - p_0) < \beta \ \&\& \ Abs(q_1 - q_0) < \beta) \quad (8\text{-}470)$$

</td></tr>
</table>

Claim 6	
defining a visual perception threshold for each of said details in accordance with said estimated detail parameters;	– If chromaEdgeFlag is equal to 0, Pg. 202 https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-H.264-200711-S!!PDF-E&type=items $\boxed{\alpha} = \alpha' * (1 << (\boxed{\mathrm{BitDepth}_Y} - 8))$ $\boxed{\beta} = \beta' * (1 << (\boxed{\mathrm{BitDepth}_Y} - 8))$ – Otherwise (chromaEdgeFlag is equal to 1), $\boxed{\alpha} = \alpha' * (1 << (\boxed{\mathrm{BitDepth}_C} - 8))$ $\boxed{\beta} = \beta' * (1 << (\boxed{\mathrm{BitDepth}_C} - 8))$ Pg. 67-68 **bit_depth_luma_minus8** specifies the bit depth of the samples of the luma array and the value of the luma quantisation parameter range offset $\mathrm{QpBdOffset}_Y$, as specified by $\boxed{\mathrm{BitDepth}_Y = 8 + \mathrm{bit\_depth\_luma\_minus8}}$ (7-2) $\mathrm{QpBdOffset}_Y = 6 * \mathrm{bit\_depth\_luma\_minus8}$ (7-3) When bit_depth_luma_minus8 is not present, it shall be inferred to be equal to 0. bit_depth_luma_minus8 shall be in the range of 0 to 6, inclusive. **bit_depth_chroma_minus8** specifies the bit depth of the samples of the chroma arrays and the value of the chroma quantisation parameter range offset $\mathrm{QpBdOffset}_C$, as specified by $\boxed{\mathrm{BitDepth}_C = 8 + \mathrm{bit\_depth\_chroma\_minus8}}$ (7-4) $\mathrm{QpBdOffset}_C = 6 * (\mathrm{bit\_depth\_chroma\_minus8} + \mathrm{residual\_colour\_transform\_flag})$ (7-5) When bit_depth_chroma_minus8 is not present, it shall be inferred to be equal to 0. bit_depth_chroma_minus8 shall be in the range of 0 to 6, inclusive.

Preliminary Claim Chart Showing Infringement of Claim 6 of the U.S. Patent No. 6,473,532 by Lenovo

Claim 6	
classifying said frame picture details into subclasses in accordance with said visual perception thresholds and said detail parameters; and	**G.8.7.4.2 SVC filtering process for a set of samples across a horizontal or vertical block edge ...** Depending on the variable filterSamplesFlag, the following applies: Pg. 498-499 – If filterSamplesFlag is equal to 1, the following applies: ***{Note: First subclass}*** – If bS is less than 4, the process specified in clause 8.7.2.3 is invoked with p_i and q_i ($i = 0..2$), chromaEdgeFlag, bS, β, and indexA given as input, and the output is assigned to p'_i and q'_i ($i = 0..2$). – Otherwise (bS is equal to 4), the process specified in clause 8.7.2.4 is invoked with p_i and q_i ($i = 0..3$), chromaEdgeFlag, α, and β given as input, and the output is assigned to p'_i and q'_i ($i = 0..2$). – Otherwise (filterSamplesFlag is equal to 0), the filtered result samples p'_i and q'_i ($i = 0..2$) are replaced by the corresponding input samples p_i and q_i: ***{Note: Second subclass}*** for $i = 0..2$, $p'_i = p_i$ (G-357) for $i = 0..2$, $q'_i = q_i$ (G-358) The variable filterSamplesFlag is derived by: filterSamplesFlag = (bS != 0 && Abs($p_0 - q_0$) < α && Abs($p_1 - p_0$) < β && Abs($q_1 - q_0$) < β) (8-460) – If chromaEdgeFlag is equal to 0, α = α′ * (1 << ($BitDepth_Y$ − 8)) β = β′ * (1 << ($BitDepth_Y$ − 8)) Pg. 202 – Otherwise (chromaEdgeFlag is equal to 1), α = α′ * (1 << ($BitDepth_C$ − 8)) β = β′ * (1 << ($BitDepth_C$ − 8)) https://www.itu.int/rec/dologin_pub.asp?lang=e&id=T-REC-H.264-200711-S!!PDF-E&type=items

Claim 6	
transforming each said frame detail in accordance with its associate subclass.	**8.3.2.2 Intra_8x8 sample prediction** Pg. 129 This process is invoked for each 8x8 luma block of a macroblock with prediction mode equal to Intra_8x8 followed by the transform decoding process and picture construction process prior to deblocking for each 8x8 luma block. … Pg. 133 **8.3.2.2.2 Specification of Intra_8x8_Vertical prediction mode** This Intra_8x8 prediction mode is invoked when Intra8x8PredMode[luma8x8BlkIdx] is equal to 0. This mode shall be used only when the samples p[x, -1] with x = 0..7 are marked as "available for Intra_8x8 prediction". The values of the prediction samples $pred8x8_L$[x, y], with x, y = 0..7 are derived by $pred8x8_L$[x, y] = p'[x, -1], with x, y = 0..7 (8-91) **8.3.2.2.3 Specification of Intra_8x8_Horizontal prediction mode** This Intra_8x8 prediction mode is invoked when Intra8x8PredMode[luma8x8BlkIdx] is equal to 1. This mode shall be used only when the samples p[-1, y], with y = 0..7 are marked as "available for Intra_8x8 prediction". The values of the prediction samples $pred8x8_L$[x, y], with x, y = 0..7 are derived by $pred8x8_L$[x, y] = p'[-1, y], with x, y = 0..7 (8-92) … ***{Note: subclasses for p' and q' output in slide 11 are used in many additional transformations of frame details }***